

Elliptic Integrals

William H. Press, and Saul A. Teukolsky

Citation: [Computers in Physics](#) **4**, 92 (1990); doi: 10.1063/1.4822893

View online: <https://doi.org/10.1063/1.4822893>

View Table of Contents: <http://aip.scitation.org/toc/cip/4/1>

Published by the [American Institute of Physics](#)

Articles you may be interested in

[Elliptic integrals and functions](#)

[Computers in Physics](#) **9**, 268 (1995); 10.1063/1.168529

Elliptic Integrals

William H. Press and Saul A. Teukolsky

Elliptic integrals occur in many applications because any integral of the form

$$\int R(t,s)dt, \quad (1)$$

where R is a rational function of t and s , and s is the square root of a cubic or quartic polynomial in t , can be evaluated in terms of elliptic integrals. Standard references¹ describe how to carry out the reduction, which was originally done by Legendre. Legendre showed that only three basic elliptic integrals are required. The simplest of these is

$$I_1 = \int_y^x \frac{dt}{\sqrt{(a_1 + b_1 t)(a_2 + b_2 t)(a_3 + b_3 t)(a_4 + b_4 t)}}, \quad (2)$$

where we have written the quartic s^2 in factored form. In standard integral tables, one of the limits of integration must be a zero of the quartic. If neither x nor y is a zero, the integral (2) must be split in two, each part having a zero at one limit. The tables then distinguish eight cases, four zeros each being either the upper or lower limit. Moreover, the limit $b_4 \rightarrow 0$ reduces the quartic to a cubic and effectively places one of the roots at infinity; this leads to eight more cases. These 16 cases are then usually tabulated in terms of Legendre's standard elliptic integral of the 1st kind, which we will define below. By a change of the variable of integration t , the zeros of the quartic are mapped to standard locations on the real axis. Then only two dimensionless parameters are needed to tabulate Legendre's integral. However, the symmetry of the original integral (2) under permutation of the roots is concealed in Legendre's notation.

Carlson² has given a new definition of a standard elliptic integral of the first kind,

$$R_F(x,y,z) = \frac{1}{2} \int_0^\infty \frac{dt}{\sqrt{(t+x)(t+y)(t+z)}}, \quad (3)$$

where x , y , and z are nonnegative and at most one is zero. By standardizing the range of integration, he retains permutation symmetry for the zeros. (Weierstrass' canonical form also has this property.) Carlson first shows that when x or y is a zero of the quartic in (2), the integral I_1 can be written in terms of R_F in a form that is symmetric under permutation of the remaining three zeros. In the general case when neither x nor y is a zero, two such R_F functions can be combined into a single one

by an *addition theorem*, leading to the fundamental formula

$$I_1 = 2R_F(U_{12}^2, U_{13}^2, U_{14}^2), \quad (4)$$

where

$$U_{ij} = (X_i X_j Y_k Y_m + Y_i Y_j X_k X_m)/(x - y), \quad (5)$$

$$X_i = (a_i + b_i x)^{1/2}, \quad Y_i = (a_i + b_i y)^{1/2}, \quad (6)$$

and ij,k,m is a permutation of 1, 2, 3, 4. A shortcut in evaluating these expressions is

$$\begin{aligned} U_{13}^2 &= U_{12}^2 - (a_1 b_4 - a_4 b_1)(a_2 b_3 - a_3 b_2), \\ U_{14}^2 &= U_{12}^2 - (a_1 b_3 - a_3 b_1)(a_2 b_4 - a_4 b_2). \end{aligned} \quad (7)$$

The U 's correspond to the three ways of pairing the four zeros, and I_1 is thus manifestly symmetric under permutation of the zeros. Equation (4) therefore reproduces all 16 cases when one limit is a zero, and also includes the cases when neither limit is zero.

Thus Carlson's function allows arbitrary ranges of integration and arbitrary positions of the branch points of the integral relative to the interval of integration. To handle elliptic integrals of the second and third kind, Carlson defines the standard integral of the third kind as

$$R_J(x,y,z,p) = \frac{3}{2} \int_0^\infty \frac{dt}{(t+p)\sqrt{(t+x)(t+y)(t+z)}}, \quad (8)$$

which is symmetric in x , y , and z . The degenerate case when two arguments are equal is denoted

$$R_D(x,y,z) = R_J(x,y,z,z) \quad (9)$$

and is symmetric in x and y . The function R_D replaces Legendre's integral of the second kind. The degenerate form of R_F is denoted

$$R_C(x,y) = R_F(x,y,y). \quad (10)$$

It embraces logarithmic, inverse circular, and inverse hyperbolic functions.

Carlson^{3,4} gives integral tables in terms of the exponents of the linear factors of the quartic in (1). For example, the integral where the exponents are $(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})$ can be expressed as a single integral in terms of R_D ; it accounts for 144 separate cases in Gradshteyn and Ryzhik!⁵

Suppose you really want to express your answer in Legendre's notation; or suppose you need the numerical values of some elliptic integrals given to you in that notation and you plan to use the numerical algorithms

William H. Press is a professor of astronomy and physics at Harvard University. Saul A. Teukolsky is a professor of physics and astronomy at Cornell University.

given below. It is a simple matter to transform back and forth. The *Legendre elliptic integral of the 1st kind* is defined as

$$F(\phi, k) \equiv \int_0^\phi \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}}. \quad (11)$$

The *complete elliptic integral of the 1st kind* is given by

$$K(k) \equiv F(\pi/2, k). \quad (12)$$

In terms of R_F ,

$$\begin{aligned} F(\phi, k) &= \sin \phi R_F(\cos^2 \phi, 1 - k^2 \sin^2 \phi, 1), \\ K(k) &= R_F(0, 1 - k^2, 1). \end{aligned} \quad (13)$$

The *Legendre elliptic integral of the 2nd kind* and the *complete elliptic integral of the 2nd kind* are given by

$$\begin{aligned} E(\phi, k) &\equiv \int_0^\phi \sqrt{1 - k^2 \sin^2 \theta} d\theta \\ &= \sin \phi R_F(\cos^2 \phi, 1 - k^2 \sin^2 \phi, 1) \\ &\quad - \frac{1}{3} k^2 \sin^3 \phi R_D(\cos^2 \phi, 1 - k^2 \sin^2 \phi, 1), \\ E(k) &\equiv E(\pi/2, k) \\ &= R_F(0, 1 - k^2, 1) - \frac{1}{3} k^2 R_D(0, 1 - k^2, 1). \end{aligned} \quad (14)$$

Finally, the *Legendre elliptic integral of the 3rd kind* is

$$\begin{aligned} \Pi(\phi, n, k) &\equiv \int_0^\phi \frac{d\theta}{(1 + n \sin^2 \theta) \sqrt{1 - k^2 \sin^2 \theta}} \\ &= \sin \phi R_F(\cos^2 \phi, 1 - k^2 \sin^2 \phi, 1) - \frac{1}{3} n \sin^3 \phi \\ &\quad \times R_J(\cos^2 \phi, 1 - k^2 \sin^2 \phi, 1 + n \sin^2 \phi). \end{aligned} \quad (15)$$

(Note that this sign convention for n is opposite that of Abramowitz and Stegun,⁶ and that their $\sin \alpha$ is our k .)

Carlson's functions are homogeneous of degree $-\frac{1}{2}$ and $-\frac{3}{2}$, so

$$\begin{aligned} R_F(\lambda x, \lambda y, \lambda z) &= \lambda^{-1/2} R_F(x, y, z), \\ R_J(\lambda x, \lambda y, \lambda z, \lambda p) &= \lambda^{-3/2} R_J(x, y, z, p). \end{aligned} \quad (16)$$

Thus, to express a Carlson function in Legendre's notation, permute the arguments into ascending order, use homogeneity to scale the third argument to be 1, and then use Eqs. (13)–(15).

In some applications the argument p in R_J or the argument y in R_C is negative, and the Cauchy principal value of the integral is required. This is easily handled by using the formulas

$$\begin{aligned} R_J(x, y, z, p) &= [(\gamma - y) R_J(x, y, z, \gamma) - 3 R_F(x, y, z) \\ &\quad + 3 R_C(xz/y, p\gamma/y)] / (y - p), \end{aligned} \quad (17)$$

where

Box 1. A Note on FORTRAN Style

Regular readers of this column will notice some changes in our FORTRAN style starting with this issue. Briefly, the idea is to bring the subroutine presentation closer to convergence with an accepted standard of good programming practice in other languages.

From now on, we plan to declare all variables and identifiers in "type" statements (REAL, INTEGER, etc.), whether or not the declarations are required by the language. Not only will this allow the routines to be translated more easily from FORTRAN to other languages, but also will simplify the task of a user who wants to convert them to double precision.

We further plan to separate, typographically at least, a routine's "public" or "interface" section from its "private" or "implementation" section. We do this to be more concordant with modular and object-oriented programming styles—although FORTRAN is by no means a modular or object-oriented language! Our *public* section contains the calling interface and declaration of its variables. We find it useful to consider PARAMETER statements, and their associated declarations, as also being in the public section, since a user may want to modify parameter values to suit a particular purpose. COMMON blocks are also a part of the public section, since they involve communication between routines. Last in the public section will be a standardized comment with the word "USES" followed by a list of all external subroutines and functions that the routine references (excluding built-in FORTRAN functions).

Our familiar principal comment, explaining briefly the use of the routine, will separate the public from the private sections. The *private* section will contain the declarations of variables that are used only internally in the routine, any necessary SAVE statements for these variables, and (of course) the routine's actual executable code.

Another change is in our use of upper and lower case. Case is not significant to the language, so it can be used to promote readability. Our convention is to use uppercase for two different, nonconflicting, purposes. First, nonexecutable compiler keywords are in upper case (e.g., SUBROUTINE, REAL, COMMON); second, parameter identifiers are in upper case. The reason for capitalizing parameters is that, because their values are liable to be modified, the user often needs to scan the implementation section of code to see exactly how the parameters are used.

Finally, we are going to use the "do...enddo" construction in do-loops. While this is not a part of the FORTRAN-77 standard, its use is now so widespread as to make it a *de facto* standard—and it is included in the new FORTRAN-8X standard. Microsoft FORTRAN Version 5.0 implements "do...enddo" on PCs; it has been part of VAX FORTRAN for a decade. For the benefit of users with only minimally conforming FORTRAN-77 compilers, we will provide tiny suggested statement numbers, matched between the "do" and the "enddo," to facilitate conversion to "do...continue" format. (There are no examples of "do...enddo" in this month's routines.)

This format, or some variant of it, will be adopted in a future edition of the *Numerical Recipes* books. We welcome reader feedback and suggestions while there is still time to make changes in the adopted format.

Box 2.

```

FUNCTION rf(x,y,z)
REAL rf,x,y,z,ERRTOL,TINY,BIG,THIRD,C1,C2,C3,C4
PARAMETER(ERRTOL=.08,TINY=1.5E-38,BIG=3.E37,THIRD=1./3.,
* C1=1./24.,C2=.1,C3=3./44.,C4=1./14.)
  Computes Carlson's elliptic integral of the first kind,  $R_F(x,y,z)$ .  $x$ ,  $y$  and  $z$  must be non-
  negative, and at most one can be zero. TINY must be at least 5 times the machine underflow
  limit, BIG at most one fifth the machine overflow limit.
REAL alamb,ave,delx,dely,dely,e2,e3,sqrtx,sqrty,sqrtz,xt,yt,zt
if(min(x,y,z).lt.0..or.min(x+y,x+z,y+z).lt.TINY.or.
* max(x,y,z).gt.BIG)pause 'invalid arguments in rf'
  xt=x
  yt=y
  zt=z
1 continue
  sqrtx=sqrt(xt)
  sqrty=sqrt(yt)
  sqrtz=sqrt(zt)
  alamb=sqrtx*(sqrty+sqrtz)+sqrty*sqrtz
  xt=.25*(xt+alamb)
  yt=.25*(yt+alamb)
  zt=.25*(zt+alamb)
  ave=THIRD*(xt+yt+zt)
  delx=(ave-xt)/ave
  dely=(ave-yt)/ave
  dely=(ave-zt)/ave
  if(max(abs(delx),abs(dely),abs(dely)).gt.ERRTOL)go to 1
  e2=delx*dely-dely**2
  e3=delx*dely*dely
  rf=(1.+(C1*e2-C2-C3*e3)*e2+C4*e3)/sqrt(ave)
  return
END

```

$$\gamma \equiv y + (z - y)(y - x)/(y - p) \quad (18)$$

is positive if p is negative, and

$$R_C(x,y) = [x/(x-y)]^{1/2} R_C(x-y, -y). \quad (19)$$

The Cauchy principal value of R_J has a zero at some value of $p < 0$, so (17) will give some loss of significant figures near the zero.

We refer you to Carlson's papers²⁻⁴ for some of the practical details in reducing elliptic integrals to his standard forms, such as handling complex conjugate zeros.

Turn now to the numerical evaluation of elliptic integrals. The traditional methods⁷⁻⁹ are Gauss or Landen transformations. *Descending* transformations decrease the modulus k of the Legendre integrals toward zero, *increasing* transformations increase it toward unity. In these limits the functions have simple analytic expressions. While these methods converge quadratically and are quite satisfactory for integrals of the first and second kinds, they generally lead to loss of significant figures in certain regimes for integrals of the third kind. Carlson's algorithms,^{10,11} by contrast, provide a unified method for all three kinds with no significant cancellations.

The key ingredient in these algorithms is the *duplication theorem*:

$$R_F(x,y,z) = 2R_F(x+\lambda,y+\lambda,z+\lambda) \\ = R_F\left(\frac{x+\lambda}{4}, \frac{y+\lambda}{4}, \frac{z+\lambda}{4}\right), \quad (20)$$

where

$$\lambda = (xy)^{1/2} + (xz)^{1/2} + (yz)^{1/2}. \quad (21)$$

This theorem can be proved by a simple change of variable of integration.¹² Equation (20) is iterated until the arguments of R_F are nearly equal. For equal arguments we have

Box 3.

```

FUNCTION rj(x,y,z,p)
REAL rj,p,x,y,z,xc,rf,ERRTOL,TINY,BIG,C1,C2,C3,C4,C5,C6,C7,C8
PARAMETER(ERRTOL=.05,TINY=2.5E-13,BIG=9.E11,C1=3./14.,C2=1./3.,
* C3=3./22.,C4=3./26.,C5=.75*C3,C6=1.5*C4,C7=.5*C2,C8=C3+C3)
C USES rc,rf
  Computes Carlson's elliptic integral of the third kind,  $R_J(x,y,z,p)$ .  $x$ ,  $y$  and  $z$  must be
  nonnegative, and at most one can be zero.  $p$  must be nonzero. If  $p < 0$ , the Cauchy principal
  value is returned. TINY must be at least twice the cube root of the machine underflow limit,
  BIG at most one fifth the cube root of the machine overflow limit.
REAL a,alamb,alpha,ave,b,beta,delp,dely,dely,ea,eb,ec,ed,ee,
* fac,pt,rcx,rho,sqrtx,sqrty,sqrtz,sum,tau,xt,yt,zt
if(min(x,y,z).lt.0..or.min(x+y,x+z,y+z,abs(p)).lt.TINY.or.
* max(x,y,z,abs(p)).gt.BIG)pause 'invalid arguments in rj'
  sum=0.
  fac=1.
  if(p.gt.0.)then
    xt=x
    yt=y
    zt=z
    pt=p
  else
    xt=min(x,y,z)
    zt=max(x,y,z)
    yt=x+y-z-xt-zt
    a=1./(yt-pt)
    b=a*(zt-yt)*(yt-xt)
    pt=yt+b
    rho=xt*zt/yt
    tau=pt/yt
    rcx=rc(rho,tau)
  endif
1 continue
  sqrtx=sqrt(xt)
  sqrty=sqrt(yt)
  sqrtz=sqrt(zt)
  alamb=sqrtx*(sqrty+sqrtz)+sqrty*sqrtz
  alpha=(pt*(sqrtx+sqrty+sqrtz)+sqrtx*sqrty*sqrtz)**2
  beta=pt*(pt+alamb)**2
  sum=sum+fac*rc(alpha,beta)
  fac=.25*fac
  xt=.25*(xt+alamb)
  yt=.25*(yt+alamb)
  zt=.25*(zt+alamb)
  pt=.25*(pt+alamb)
  ave=.2*(xt+yt+zt+pt+pt)
  delx=(ave-xt)/ave
  dely=(ave-yt)/ave
  dely=(ave-zt)/ave
  delp=(ave-pt)/ave
  if(max(abs(delx),abs(dely),abs(dely),abs(delp)).gt.ERRTOL)go to 1
  ea=delx*(dely+dely)+dely*dely
  eb=delx*dely*dely
  ec=delp**2
  ed=ea-3.*ec
  ee=eb+2.*delp*(ea-ec)
  rj=3.*sum+fac*(1.+ed*(-C1+C5*ed-C6*ee)+eb*(C7+delp*(-C8+delp*C4))
  +delp*ea*(C2-delp*C3)-C2*delp*ec)/(ave*sqrt(ave))
  if (p.le.0.) rj=a*(b*rj+3.*(rcx-rf(xt,yt,zt)))
  return
END

```

$$R_F(x,x,x) = x^{-1/2}. \quad (22)$$

When arguments are close enough, the function is evaluated from a fixed Taylor expansion about (22) through fifth-order terms. While the iterative part of the algorithm is only linearly convergent, the error ultimately decreases by a factor of $4^6 = 4096$ for each iteration. Typically only two or three iterations are required, perhaps six or seven if the initial values of the arguments have huge ratios. We list the algorithm for R_F here, and refer you to Carlson's paper¹⁰ for the other cases.

• For $n = 0, 1, 2, \dots$ compute

$$\mu_n = (x_n + y_n + z_n)/3,$$

$$X_n = 1 - \left(\frac{x_n}{\mu_n}\right), \quad Y_n = 1 - \left(\frac{y_n}{\mu_n}\right),$$

Box 4.

```

FUNCTION rd(x,y,z)
REAL rd,x,y,z,ERRTOL,TINY,BIG,C1,C2,C3,C4,C5,C6
PARAMETER(ERRTOL=.05,TINY=1.E-25,BIG=4.5E21,C1=3./14.,C2=1./6.,
*      C3=9./22.,C4=3./28.,C5=.25*C3,C6=1.5*C4)
*      Computes Carlson's elliptic integral of the second kind,  $R_D(z,y,z)$ .  $z$  and  $y$  must be non-
negative, and at most one can be zero.  $z$  must be positive. TINY must be at least twice the
negative 2/3 power of the machine overflow limit. BIG must be at most  $0.1 \times \text{ERRTOL}$  times
the negative 2/3 power of the machine underflow limit.
REAL alamb,ave,delx,dely,deltz,ea,eb,ec,ed,ee,fac,sqrtx,sqrtz,
*      sqrtz,sum,xt,yt,zt
if(min(x,y).lt.0..or.min(x+y,z).lt.TINY.or.
*      max(x,y,z).gt.BIG)pause 'invalid arguments in rd'
xt=x
yt=y
zt=z
sum=0.
fac=1.
1 continue
sqrtx=sqrt(xt)
sqrty=sqrt(yt)
sqrtz=sqrt(zt)
alamb=sqrtx*(sqrty+sqrtz)+sqrty*sqrtz
sum=sum+fac/(sqrtz*(zt+alamb))
fac=.25*fac
xt=.25*(xt+alamb)
yt=.25*(yt+alamb)
zt=.25*(zt+alamb)
ave=.2*(xt+yt+3.*zt)
delx=(ave-xt)/ave
dely=(ave-yt)/ave
deltz=(ave-zt)/ave
if(max(abs(delx),abs(dely),abs(deltz)).gt.ERRTOL)go to 1
ea=delx*dely
eb=deltz*dely
ec=ea*eb
ed=ea*.6.*eb
ee=ed*ec*ec
rd=3.*sum+fac*(1.+ed*(-C1+C5*ed-C6*deltz*ee)
*      +deltz*(C2*ee+deltz*(-C3*ec+deltz*C4*ea)))/(ave*sqrt(ave))
return
END

```

$$Z_n = 1 - \left(\frac{z_n}{\mu_n} \right),$$

$$\epsilon_n = \max(|X_n|, |Y_n|, |Z_n|).$$

If $\epsilon_n < \text{tol}$ go to next stage; else compute

Box 5.

```

FUNCTION rc(x,y)
REAL rc,x,y,ERRTOL,TINY,SQRTNY,BIG,TNMG,COMP1,COMP2,THIRD,
*      C1,C2,C3,C4
*      PARAMETER(ERRTOL=.04,TINY=1.69E-38,SQRTNY=1.3E-19,BIG=3.E37,
*      TNMG=TINY*BIG,COMP1=2.236/SQRTNY,COMP2=TNMG*TNMG/25.,
*      THIRD=1./3.,C1=.3,C2=1./7.,C3=.375,C4=9./22.)
*      Computes Carlson's degenerate elliptic integral,  $R_C(z,y)$ .  $z$  must be nonnegative and  $y$  must
be nonzero. If  $y < 0$ , the Cauchy principal value is returned. TINY must be at least 5 times
the machine underflow limit, BIG at most one fifth the machine maximum overflow limit.
REAL alamb,ave,s,v,xt,yt
if(x.lt.0..or.y.eq.0..or.(x*abs(y)).lt.TINY.or.(x*abs(y)).gt.BIG
*      .or.(y.lt.-COMP1.and.x.gt.0..and.x.lt.COMP2))
*      pause 'invalid arguments in rc'
if(y.gt.0.)then
xt=x
yt=y
w=1.
else
xt=x-y
yt=-y
v=sqrt(x)/sqrt(xt)
endif
1 continue
alamb=2.*sqrt(xt)*sqrt(yt)+yt
xt=.25*(xt+alamb)
yt=.25*(yt+alamb)
ave=THIRD*(xt+yt+yt)
s=(yt-ave)/ave
if(abs(s).gt.ERRTOL)go to 1
rc=w*(1.+s*s*(C1+s*(C2+s*(C3+s*C4))))/sqrt(ave)
return
END

```

$$\lambda_n = (x_n y_n)^{1/2} + (x_n z_n)^{1/2} + (y_n z_n)^{1/2},$$

$$x_{n+1} = \frac{(x_n + \lambda_n)}{4}, \quad y_{n+1} = \frac{(y_n + \lambda_n)}{4},$$

$$z_{n+1} = \frac{(z_n + \lambda_n)}{4},$$

and repeat this stage.

• Compute

$$E_2 = X_n Y_n - Z_n^2, \quad E_3 = X_n Y_n Z_n,$$

$$R_F = (1 - \frac{1}{10}E_2 + \frac{1}{14}E_3 + \frac{1}{24}E_2^2 - \frac{3}{44}E_2 E_3)/(\mu_n)^{1/2}.$$

A value of 0.08 for the error tolerance parameter is adequate for single precision (seven significant digits). Since the error scales as ϵ_n^6 , we see that 0.0025 will yield double precision (16 significant digits) and require at most two or three more iterations. Since the coefficients of the sixth-order truncation error are different for the other elliptic functions, these values for the error tolerance should be changed to 0.04 and 0.0012 in the algorithm for R_C , and 0.05 and 0.0015 for R_J and R_D . As well as being an algorithm in its own right for certain combinations of elementary functions, the algorithm for R_C is used repeatedly in the computation of R_J .

The FORTRAN implementations test the input arguments against two machine-dependent constants, TINY and BIG, to ensure that there will be no underflow or overflow during the computation. We have chosen conservative values, corresponding to a machine minimum of 3×10^{-39} and a machine maximum of 1.7×10^{38} . You can always extend the range of admissible argument values by using the homogeneity relations (16). ■

In our next column: Adaptive allocation of sample points in Monte Carlo integration.

References

1. A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi, *Higher Transcendental Functions* (McGraw-Hill, New York, 1953), Vol. II.
2. B. C. Carlson, *SIAM J. Math. Anal.* **8**, 231 (1977).
3. B. C. Carlson, *Math. Comput.* **49**, 595 (1987).
4. B. C. Carlson, *Math. Comput.* **51**, 267 (1988).
5. I. S. Gradshteyn and I. W. Ryzhik, *Table of Integrals, Series, and Products* (Academic, New York, 1980).
6. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (National Bureau of Standards, Washington, DC, 1964).
7. For example, R. Bulirsch, *Num. Math.* **7**, 78 (1965); **7**, 353 (1965); **13**, 305 (1969).
8. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge U. P., New York, 1986).
9. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge U. P., New York, 1988).
10. B. C. Carlson, *Num. Math.* **33**, 1 (1979).
11. B. C. Carlson and E. M. Notis, *ACM Trans. Math. Software* **7**, 398 (1981).
12. B. C. Carlson, *SIAM J. Math. Anal.* **9**, 524 (1978).